INTERPRETABILITY OF DEEP LEARNING MODELS

A Project Report

submitted by

SAHANA RAMNATH

in partial fulfilment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY
&
MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING &
DEPARTMENT OF COMPUTER SCIENCE
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

JUNE 2020

THESIS CERTIFICATE

This is to certify that the thesis titled **INTERPRETABILITY OF DEEP LEARNING MODELS**, submitted by **SAHANA RAMNATH (EE15B109)**, to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelor of Technology and Master of Technology**, is a bonafide record of the research work done by her under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Mitesh M. Khapra
Research Guide
Professor
Dept. of Computer Science
IIT Madras, 600 036

Place: Chennai

Date: June 18, 2020

ACKNOWLEDGEMENTS

First, I would like to express my sincere gratitude to my DDP advisor, Prof. Mitesh Khapra, for his constant guidance and encouragement throughout the course of my project. I would like to thank Prof. Mitesh Khapra and Prof. Ravindran Balaraman for providing the lab GPU cluster resources to run my experiments. I would like to sincerely thank Preksha Nema, research scholar under Prof. Khapra, for her extensive guidance and help in ideating and analyzing experiments. I would like to thank my lab-mate Deep Sahni for assisting me with experiments and analysis of results in this project. I would also like to thank Amrita Saha(IBM-IRL) and Prof. Soumen Chakrabarti(IITB) for guiding me through the Image Retrieval Project. Last but not least, I would like to profoundly thank my parents, sister, and friends for supporting and encouraging my education and research.

ABSTRACT

The past few years have seen an explosion in the development of artificial intelligence systems to tackle a plethora of human tasks, starting with simple tasks such as image classification, and going on to more complex ones such as answering questions based on reading passages or images, language translation, or playing games such as AlphaGo or Minecraft. With the increasing size and complexity of deep learning models, accuracy on various tasks has increased tremendously(almost human-level); it is now necessary to take a step back and analyze whether these models are working in an explainable and interpretable way. This dual degree project consists of two concurrent threads, with 'Interpretability of Deep Learning Systems' as the central theme: Analyzing Interpretability of Deep RCQA Systems and Dialog-Based Image Retrieval.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The first thread of this project is the interpretability analysis of existing deep models for the task of Reading-Comprehension based Question Answering(RCQA) on the dataset SQuAD (Rajpurkar *et al.*, 2016). Rapid progress in RCQA over the past few years has led to the development of increasingly complex neural architectures with specialized layers and components for modeling various aspects of QA. SQuAD is one of the most commonly used datasets for the task of RC based question-answering; extensive work has been done on this dataset, and it has been solved almost completely, with the current state-of-the-art 92.4% F1 (ALBERT (Lan *et al.*, 2019)) beating the human score of 89.45% F1. The dataset consists of a reading passage of around 100-300 words and a natural language question, which is answered by a span in the passage itself. In this thesis 4 published models are analyzed - BiDAF (Seo *et al.*, 2016*a*), DCN (Xiong *et al.*, 2016*a*), QANet (Yu *et al.*, 2018*b*) and BERT (Devlin *et al.*, 2018*b*) on the basis of how interpretable they are when tested on SQuAD.

The second thread of this project is the implementation of a novel, explainable dialog-based image retrieval system on the CLEVR Dialog dataset (Kottur *et al.*, 2019). Most state-of-the-art techniques model this task of text-based image retrieval in a purely neural way, making it almost impossible to incorporate pragmatic strategies in retrieving images from a large-scale catalog. A novel neural-symbolic model is implemented for this task; to facilitate this, the dialog, the catalog of images, and each image are represented in terms of their scene-graphs. The task is modeled as a graph matching problem, trained end-to-end with the REINFORCE (Sutton *et al.*, 2000) algorithm. In this thesis, the complete model and its results are presented for the CLEVR-Dialog task.

CHAPTER 2

ANALYZING INTERPRETABILITY OF DEEP RCQA SYSTEMS

2.1 Motivation

Deep learning models have come close to achieving human-level accuracies on many RCQA datasets.

Dataset	Human Performance	SOTA
SQuAD	89.45%	92.42%
RACE	94.5%	89.4%
WikiHop	85.0%	78.3%
HotpotQA	91.4%	82.19%

Table 2.1: Performance on RCQA Tasks

Each new model comes with a new module designed for specific functionalities; for example, co-attention modules to understand the query and the passage together, or the self-attention module for a repeated reading of the passage. While these modules were each created for a particular purpose, there has to be a thorough analysis done to see if they are doing the work they were intended to do. In the past, the most common way to do this was to analyze the attention scores given by these modules; however, not all modules output attention scores and attention need not necessarily be the correct way of analysis. Contemporary works define 'interpretability' in different ways, using attribution based methods and further statistical analysis to quantify it across datapoints. With the massive progress in Deep Learning based NLP over the past few years, and the growing interest in this field, it is essential to analyze the work done so far, to see if we are progressing in the right direction and to promote the development of interpretable models which can be extended to real-life tasks.

2.2 Preliminaries

2.2.1 Dataset and Models Analysed

The dataset used for this analysis is **SQuAD**. With over 60 different models submitted on the leaderboard, and with the most recent models beating the human performance, SQuAD is perfectly suited for this analysis of interpretability.

First, a generic framework is introduced for the task of RCQA (refer Figure 2.1) which outlines different layers used in QA models. All layers may not be present in all models and different models may also differ in the implementation of these layers as explained below. The input to an RCQA system is typically a passage and a question. The answer

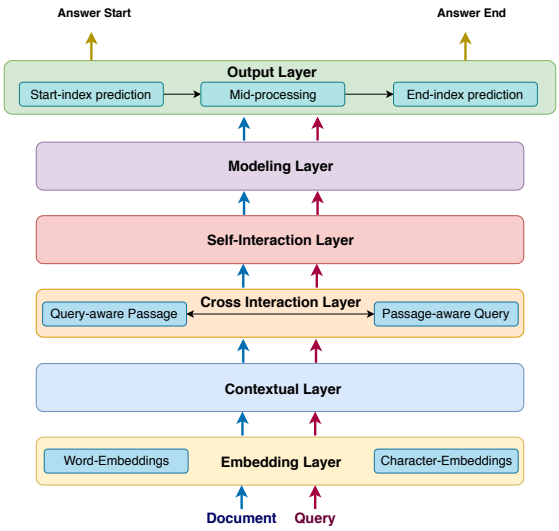


Figure 2.1: Basic Framework for RCQA

can then be (i) predicted as a span in the passage or (ii) selected from a set of given candidates or (iii) generated. In this project, only the span-based output case is focussed on. Such models contain the following layers:

i) *Word Embedding Layer*: This layer maps the passage and question words to distributed representations. These representations can be learned both at word-level and character-level.

ii) *Contextual Layer:* This layer refines the representations of each passage and question word to capture information from its neighboring words (*e.g.*, using an LSTM).

iii) *Cross-Interaction Layer:* This layer captures the interaction between the passage and question words is captured to highlight the passage words relevant to the question.

iv) *Self-Interaction Layer:* This layer helps to capture long-term associations between

passage words by allowing interactions between passage words.

v) *Modeling Layer:* This layer collates all the representations learned so far.

vi) *Output Layer:* This layer predicts the answer start and answer end, using one projection layer for each.

In this project, 4 published models - BiDAF (Seo *et al.*, 2016*b*), BERT (Devlin *et al.*, 2018*a*), DCN (Xiong *et al.*, 2016*b*) and QANet (Yu *et al.*, 2018*a*) are analysed on their performance on SQuAD. These models were re-implemented and tuned to achieve a performance close to the original F1 and EM scores on the SQuAD leaderboard.

1. **BiDAF** - This is a multi-stage hierarchical model that represents the passage at different levels of granularity(word/character) and uses a bidirectional flow mechanism between the query and the passage to obtain a query-aware passage representation. The original paper achieved an F1 score of 77.3 on the leaderboard. The reimplementation of BiDAF in this project achieved an F1 of 74.3.

2. **DCN** - This model uses a co-attentive encoder that fuses representations of the question and the passage in order to focus on relevant parts of both. It then uses a dynamic pointing decoder that iteratively alternates between estimating the start and the end of the answer span. The original paper achieved an F1 score of 75.6 on the leaderboard. The re-implementation of DCN in this project achieved an F1 of 74.11.

3. **QANet** - This model follows a similar model architecture as BiDAF and DCN. However, it replaces the RNNs using encoders that consist exclusively of convolution(local interactions) and self-attention(global interactions). The original paper achieved an F1 score of 82.7 on the leaderboard. The modified re-implementation of QANet in this project (which uses BiDAF's co-attention mechanism rather than the one mentioned in the paper) achieved an F1 of 74.9.

4. **BERT** - This is a language representational model which pre-trains deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. This allows the pre-trained BERT model (with 12 Transformer Blocks each with a multi-head self/co-attention and feed forward networks) to be used for any NLP task, by adding and fine-tuning with the task-specific output layer. The official code and pre-trained weights for BERT were released by Google Research[1], and the reported F1 score for the BERT-Base model is 88.5. In this project, BERT is not re-implemented; the official code and weights are used. Fine-tuning on the same for 2 epochs gave an F1 score of 88.73.

2.2.2 Integrated Gradients

Various techniques exist to address the task of attributing a deep network's prediction to its input features. These include LIME (Ribeiro *et al.*, 2016), DeepLift (Shrikumar

[1] https://github.com/google-research/bert

et al., 2017), Layerwise Relevance Propogation - LRP (Bach *et al.*, 2015) and Integrated Gradients (Sundararajan *et al.*, 2017). As described in Sundararajan *et al.* (2017), attribution methods must satisfy two axioms : (i) sensitivity - they should focus only on relevant features and not on irrelevant features (ii) implementation invariance - the attributions calculated by them must be identical for two functionally equivalent networks irrespective of the implementation. In this analysis, the attribution method **Integrated Gradients** is chosen, for the following reasons :

1. The motivation of this work is to completely understand the model by studying the functionality of each layer. Methods such as LIME work only with the word-level input to the model, and hence cannot be used to solve this purpose. Further, the lack of sensitivity in LIME is clearly visible when tested with deep networks such as QA models; preliminary experiments with LIME revealed that while LIME was sometimes able to highlight relevant words, it always highlights many words irrelevant to answer the question(further elaborated in Section 2.6).
2. Other gradient based attribution methods such as LRP and DeepLift replace gradients with discrete gradients and use a modified form of backpropogation. These methods depend highly on the implementation used for the discrete gradients. In addition, they require different implementations for different models, and will thus render this analysis incomparable across various models and layers.

The Integrated Gradient for a passage word w_i, embedded as $x_i \in \mathbf{R}^L$ is computed as follows:

$$IG(x_i) = \int_{\alpha=0}^{1} \frac{\partial M(\tilde{x} + \alpha(x_i - \tilde{x}))}{\partial x_i} \; d\alpha$$

where $\tilde{x}$ is a zero vector, that serves as a baseline to measure integrated gradient for w_i. In this work, the above integral is approximated across 50 uniform samples between $[0, 1]$.

2.2.3 Using integrated gradients

For a given passage D with d words $[w_1, w_2, \ldots, w_d]$, question Q, answer $[i, j]$, where i and j are start and end indices of answer span in D and a model M with ϕ parameters, the answer prediction task is modeled as:

$$p(w_s, w_e) = M(w_s, w_e | \text{embed}(D), \text{embed}(Q), \phi)$$

w_s, w_e are the predicted answer span's start and end indices. The passage and question words are embedded using embed(.), that encodes a word to $\mathbf{R}^L$ space.

For any given layer *l*, the above is equivalent to:

$$p(w_s, w_e) = f_l(w_s, w_e | E_{l-1}(P), E_{l-1}(Q), \theta)$$

where f_l is the forward propagation from layer l to the prediction. $E_l(.)$, is the representation learnt for passage or query words by a given layer *l*. To elaborate, we consider the network below the layer *l* as a blackbox which generates *input* representations for layer *l*.

The integrated gradients for each layer $IG_l(x_i)$ for all passage words w_i is calculated using Algorithm 1. Then importance scores for each w_i are computed by taking the euclidean norm of $IG(w_i)$ and then normalizing them to give a probability distribution I_l over passage words.

Algorithm 1 To compute Layer-wise Integrated Gradients for layer *l*

1: $\tilde{p} = 0$ //zero baseline
2: $m = 50$
3: $G_l(p) = \frac{1}{m} \sum_{k=1}^{m} \frac{\partial f_l(\tilde{p} + \frac{k}{m}(p - \tilde{p}))}{\partial E_l}$
4: $IG_l(p) = [(p - \tilde{p}) \times G_l(p)]$
5: // Compute squared norm at each row
6: $\tilde{I}_l([w_1, \ldots, w_k]) = ||IG_l(p)||$
7: Normalize $\tilde{I}_l$ to a probability distribution I_l

2.3 Questions Explored

The four models - BiDAF, QANet, DCN and BERT are thoroughly analyzed with different paradigms of interpretability. They are broadly analyzed on three questions - (i) Can they generate a meaningful rationale to explain their predictions? (ii) Do they progressively learn more useful/refined word representations at each layer? (iii) Do specialised layers in these QA models, such as the query-document interaction layer in BiDAF, indeed serve their intended purpose?

BERT, being the current standard model used in all NLP tasks, is further subjected to an independent analysis in an effort to understand the QA-specific role performed by each of its layers.

As described in Section 2.2.1, the different models are tuned to get a performance comparable to the reported performance in the original works, to ensure that the analysis done on these models is meaningful. The following sections describe different methods used to evaluate the model on the above questions, and corresponding results and observations.

2.4 Analysis on Extracted Rationale

With the increase in models' predictive power, it is necessary to understand how they arrive at their prediction. One line of works (Si *et al.*, 2019; Jia and Liang, 2017) analyzes models on adversarial datasets and provides insights into whether such models can understand the passage and question correctly. The others (Lei *et al.*, 2016) are trying to integrate a self-explainable component as a part of the model itself. However, creating the right adversarial dataset that could highlight (most of) the model's limitations is an expensive and difficult task. Similarly, models introduced with self-explainable components cannot be extended to existing works. In this project, a model's rationale is extracted using a simple integrated gradients framework that does not require any external inputs(such as adversarial datasets), and is extensible to all models.

The definition of a model's 'explanation' is adopted from Serrano and Smith (2019): *a model's explanation is the minimal set of items(words, in this case) which when removed collectively, causes a change in the model's prediction.*. To enumerate over the subset of passage words feasibly, the top items as ranked by integrated gradients are picked. Algorithm 2 describes how this minimal subset or *indicator* words are chosen.

Algorithm 2 Rationale Extraction

$\tilde{D}([w_1, ..., w_d]) \leftarrow ||IG(w_1)||, ..., ||IG(w_d)||$
// normalize for a probability distribution
$\tilde{D} \leftarrow \tilde{D}/\text{sum}(\tilde{D})$
$\mathbf{X} \leftarrow$ Rank $[w_1, \ldots, w_d]$ based on $\tilde{D}$
indicator_words = [ ]
repeat
    indicator_words.insert(pop($\mathbf{X}$))
until (Decision Flips)

The $IG(x_i)$ is calculated for all passage words $i \in [1, d]$ using their input embeddings. Then importance scores are computed for all w_i's by taking the euclidean norm of

$IG(w_i)$, which is then normalized to a probability distribution $\tilde{D}$. The important words are removed from the passage one by one (and replaced with **0**), until the model predicts wrong answer; i.e., when *decision flips* (Line 6-8).

2.4.1 Quantitative Analysis on Rationale Extracted

Based on the definition mentioned above for a model's explainability, it can be said that a model is *more* explainable if a smaller set of words causes its decision flip (i.e., if the rationale is *concise*). To quantify the model's explainability as per this description, consider the following measure *flip fraction*: the total fraction of words in the passage that constitute the model's rationale. Hence, a model is more explainable if the flip-fraction is lower.

These flip fractions are computed for the entire dev-split of SQuAD (10k samples). The mean and variance of these fractions across all samples are calculated and depicted in Table 2.2.

Model	Mean	Variance
BERT	0.072	0.03
BiDAF	0.161	0.064
QANet	0.165	0.065
DCN	0.215	0.059

Table 2.2: Mean and Variance of flip-fraction across SQuAD's dev set, scale of 0-1

The four models have low flip-fractions for the majority of the dataset. The low fraction values indicate that the decision flips happen as soon as the most important words as ranked by the model, are removed. From this, it can be concluded that the model is able to highlight what is important for its prediction, i.e., it is able to attribute its important features correctly (regardless of how logical this explanation is).

A potential limitation with flip-fraction is that a lower score does not necessarily ensure the quality of the rationale. Even with a low fraction, the model might be selecting wrong/insufficient words. Specifically, for the case of QA where the answer is a span from the passage, the model could directly drop the answer span itself, making it difficult to understand how exactly the model reached the answer without focusing on other necessary words.

Model	Incl. Answer Span			Excl. Answer Span		
	% P	% R	% F1	% P	% R	% F1
BERT	95.2	17.2	29.1	84.4	6.25	11.6
BiDAF	85.8	19.8	32.2	66.9	12.6	21.2
DCN	65.1	26.9	38.1	42.5	18.6	25.9
QANet	83.1	19.6	31.7	62.6	11.1	18.9

Table 2.3: Precision(P), Recall(R) and F1 score of overlap of model's rationale with human rationale

2.4.2 Human Evaluations on Rationale Extracted

To analyse the quality of the extracted rationale, human rationale is collected for 500 datapoints picked randomly and equally split amongst different question types. For the corresponding {*passage, query, answer*} triplets, two human annotators were asked to mark words and phrases in the passage, which they think are most important to arrive at the answer. They were explicitly asked not to mark *only* the answer. The two annotators were paired up and asked to resolve any discrepancies in what they individually thought was the rationale. If consensus was not reached, these points were removed. Using these human annotations as baseline, the model's rationale is checked for *precision* and *recall(completeness)*. As defined in Algorithm 2, the model's rationale is taken to be the set of indicator words at the model's input layer. The model is defined as *precise* if every word in its rationale is also present in the human-annotated rationale. It is defined as *complete* if every word in the human rationale is present in its rationale. Note that while comparing the two sets, stop words are not taken into account. The results can be found in Table 2.3.

It was found from preliminary analysis that the model often marks its predicted answer span as part of its rationale. Hence, the comparison is done as two cases: (i) including the answer span, which may be present in none/either/both of them (ii) excluding the answer span.

On analyzing the two results, it is seen that while precision is reasonably high in both cases, all the models are highly focused on the answer span itself; on excluding the answer, there is approximately a 10-20% drop in precision for all models. Further, it is observed that the recall values are much lower; this concludes that though the models seemed explainable based on flip fractions, the reasoning on how the answer was retrieved is not yet sufficient.

Question:	What was maria curie the first female recipient of ?
Answer:	Nobel Prize
DCN	One of the most famous people born in Warsaw was Maria SkÃÂodowska-Curie , who achieved international recognition for her research on radioactivity and was the first female recipient of the Nobel Prize....
BiDAF	One of the most famous people born in Warsaw was Maria SkÃÂodowska-Curie , who achieved international recognition for her research on radioactivity and was the first female recipient of the Nobel Prize....
QANET	One of the most famous people born in Warsaw was Maria SkÃÂodowska-Curie , who achieved international recognition for her research on radioactivity and was first female recipient of the Nobel Prize....
BERT	One of the most famous people born in Warsaw was Maria SkÃÂodowska-Curie , who achieved international recognition for her research on radioactivity and was the first female recipient of the Nobel Prize....

Table 2.4: Sample of Rationale Extracted for DCN, BiDAF, QANET and BERT. All the models predicted the correct answer "Nobel Prize".

.

Further, it is observed that DCN's recall (though low in an absolute sense) is the highest across the four models. This is possibly because its modeling of query-passage interaction is a simple extension of cosine similarity compared to the highly non-linear functions in other models.

Qualitative Analysis: Table 2.4 shows rationales extracted from all four models for one *{passage, question, answer}* triplet. The human rationale was "maria, curie, first female recipient of noble prize".

DCN's rationale contains necessary words such as 'maria', 'female', and 'of' required to link the question to the answer. However, it still misses out on necessary words like "first, recipient" to complete the rationale. Moreover, it highlights words like 'most famous people born' which are irrelevant to answer the question, and thus has lower precision. BiDAF and QANET, do not highlight any unnecessary words and are almost complete. BiDAF fails to highlight "Maria, first" and QANET does not highlight "first, recipient". On the other hand, BERT does not give any information in the rationale; it just selects the answer word directly. This shows that though BERT is the best-performing model on SQuAD, with the existing attribution methods, it does not provide meaningful rationales.

2.4.3 Decision Flips

Further, for the sake of completion of analysis, this section shows the decision flip fraction at each layer in the model. This experiment extends the representation erasure experiment done at the input layer to all the layers in the model. The results for each layer separately can be found in Figure 2.2.

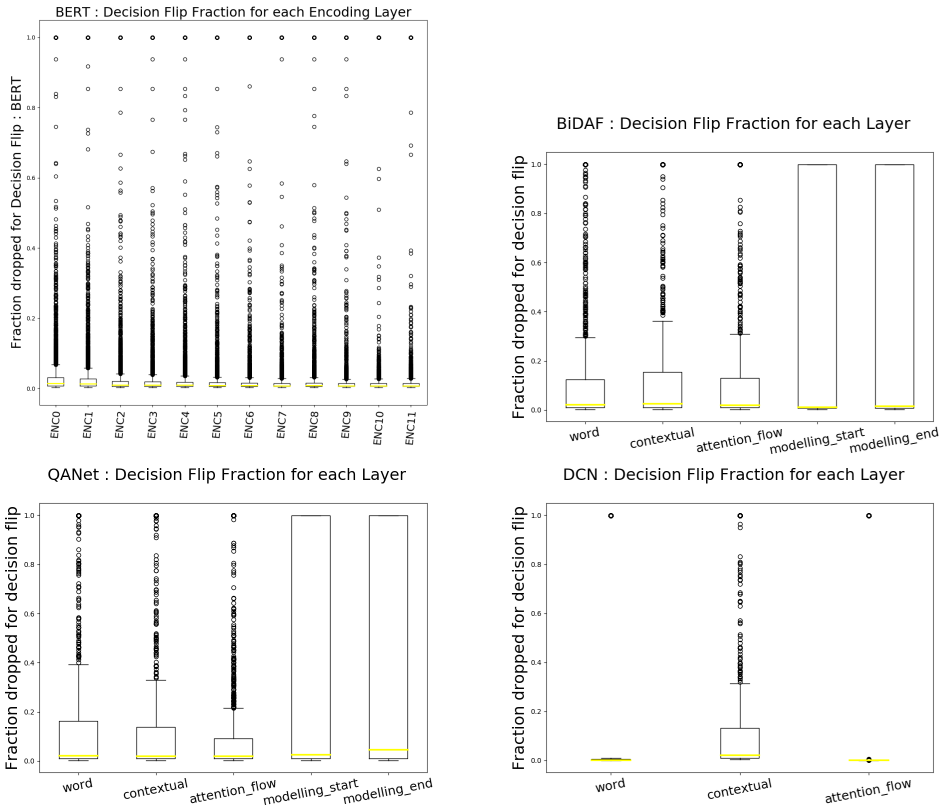


Figure 2.2: Decision Flip Fractions for BERT, BiDAF, DCN and QANet (left to right, top to bottom)

2.5 Conicity

In this section, the evolution of the word embedding vectors produced by the model at each layer is tracked and analyzed. Specifically, the metric *conicity*(Sharma *et al.*, 2018) is used to calculate the aggregated vector space similarity of these embeddings. *Conicity* between a set of vectors $\mathbf{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$ is defined as follows : First compute a vector $\mathbf{v}_i$'s ATM ('alignment to mean') as the cosine similarity between vector $\mathbf{v}_i$ and

the mean of all vectors in $\mathbf{V}$

$$\text{ATM}(\mathbf{v}_i, \mathbf{V}) = \text{cosine}(\mathbf{v}_i, \frac{1}{m}\sum_{j=1}^{m}\mathbf{v}_j)$$

Conicity of the set $\mathbf{v}_i$ is now defined as the mean ATM of all vectors in $\mathbf{V}$

$$\text{Conicity}(\mathbf{V}) = \frac{1}{m}\sum_{i=1}^{m}\text{ATM}(\mathbf{v}_i, \mathbf{V})$$

From this definition, a high value of conicity means that the vectors in $\mathbf{V}$ form a narrow cone, centered at the origin; that is, they are all highly aligned with each other.

Conicity is computed for all the word embeddings per datapoint, at each layer of the model. The resulting set of conicities across all datapoints is visualized using a histogram (refer Figure 2.3).

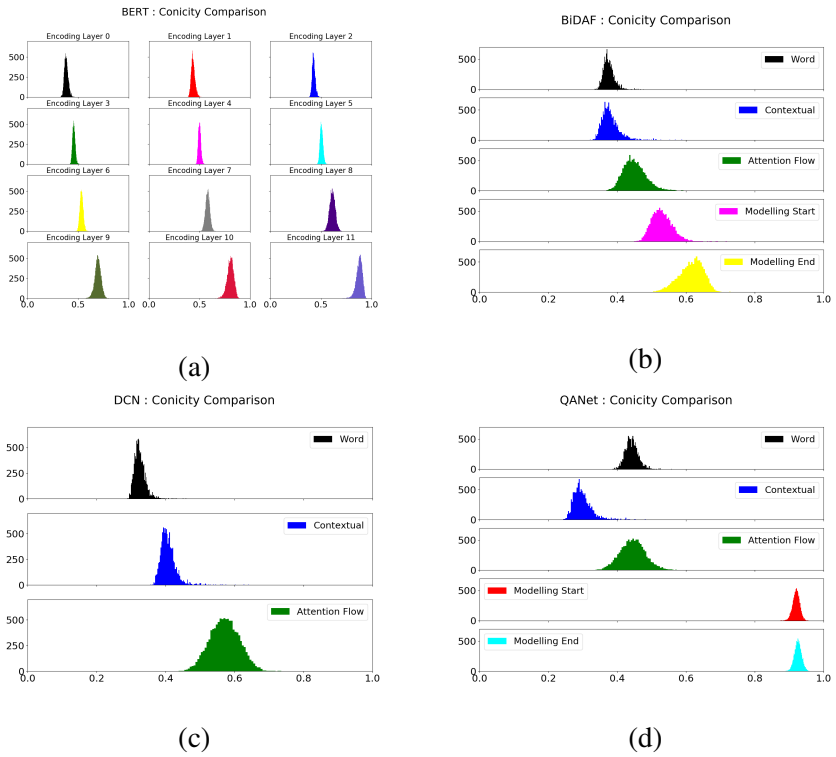


Figure 2.3: Histogram of conicity across different layers for BERT, BiDAF, DCN, QANET (left to right; top to bottom)

It is observed that for all models, the conicity calculated seems to increase as from the initial layers to the deeper layers. It might be expected that the deeper layers of the model would have a more discriminatory power, and hence, the word representations in these layers would be farther apart from each other. However, that does not seem to be

the case, and it is not clear what to infer from this increasing conicity or why should it increase. However, a limitation of this experiment is that, we can't separately view the change in conicity for the answer span words and the other words separately; it may be the case that all the words' embeddings come together, but the answer span words alone are farther away and hence distinguishable. To analyze this, conicity is now calculated for the embeddings of the same word across all the layers (this experiment is done only for BERT since in all other models, the embedding dimension changes across layers making this experiment infeasible).

Figure 2.4 shows the evolution of embeddings of the start/end of the span, the separator

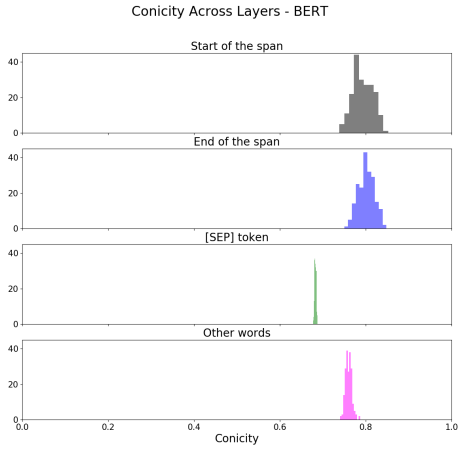


Figure 2.4: Conicity - Evolution of word embeddings across layers in BERT

token [SEP] and the other words (averaged) across the 12 layers of BERT. It is seen that the conicity is fairly high indicating high similarity between the word embeddings across the layers. A preliminary conclusion is that the model isn't doing much work across its layer and that the embedding at the final layer is very similar to the input layer. However, a deeper conclusion is that, even though the mathematical perturbations across layers is small, it is significant to the model's performance; this suggests that current interpretability methods like conicity are not sensitive enough to capture such changes.

2.6 LIME

Integrated Gradients was the attribution method chosen for this analysis. However, as a sanity check, experiments were also done using LIME, and the resulting importance

scores assigned to words was visualised in Figure 2.5. It can be seen from the that while Integrated Gradients focuses on the more relevant words with perhaps the highest focus on the answer span, LIME is placing emphasis on a lot of irrelevant words, clearly indicating that the latter doesn't satisfy the sensitivity axiom.

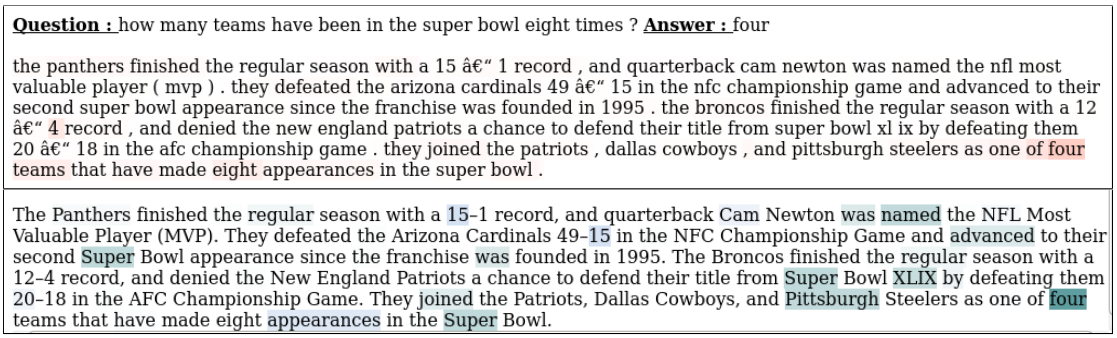


Figure 2.5: BERT : Visualisation of Importance Scores given to the Passage using Integrated Gradients(top) and LIME(bottom).

2.7 Visualization of Attention Mechanism

Each model has its own unique attention mechanism. While these do not represent the working of the entire model, they can be analyzed to understand the workings of that particular layer. As seen in Table 2.6, in BiDAF, DCN and QANet, in the query-passage co-attention layer, the question seems to be paying attention mostly to wherever it occurs in the passage. In the attention head plots in Table 2.5, it can be seen that the heads mostly show one of these properties : (i) they are positional, and each word gives importance only to itself and maybe 1/2 immediate surrounding words (ii) all words give importance to the [SEP] or [CLS] tokens (iii) they give high importance to punctuation marks alone (multiple parallel lines in the last plot).

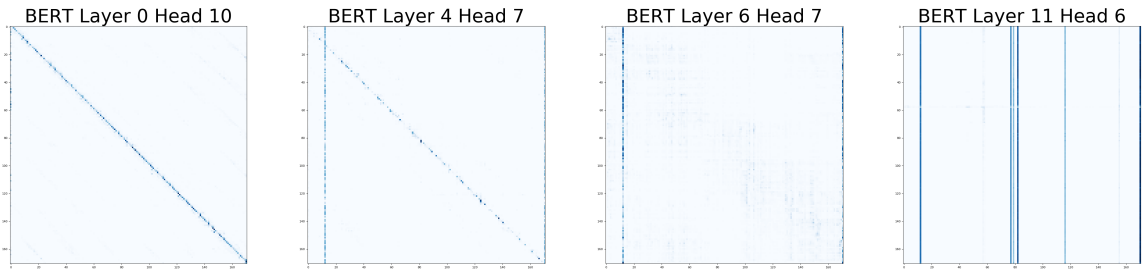


Table 2.5: Some Attention Head Visualizations from BERT

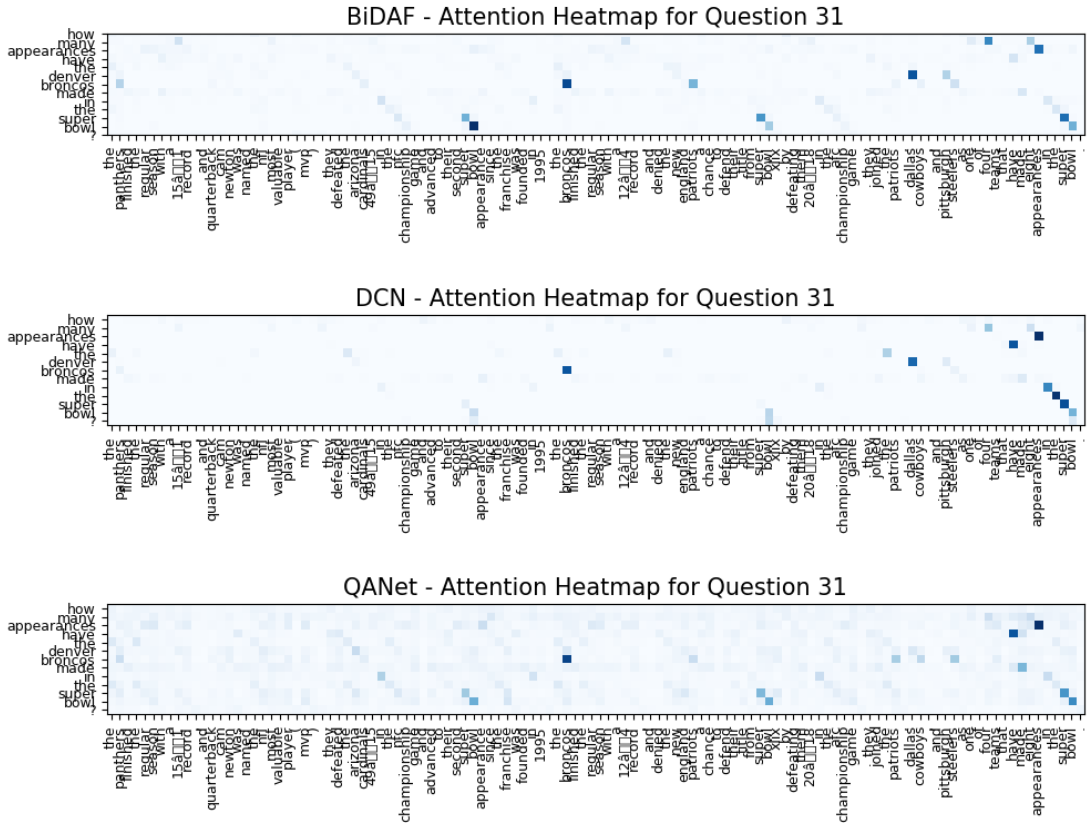


Table 2.6: Attention Mechanism Plot for BiDAF, DCN and QANet (top to bottom)

2.8 Interpreting BERT's layers

BERT and its variants have replaced SOTA performance in multiple NLP tasks. Earlier works (Tenney *et al.*, 2019; Peters *et al.*, 2018) analyze syntactic and semantic purposes for each layer in such models. Clark *et al.* (2019) specifically analyses BERT's attention heads for syntactic and linguistic phenomena. These works focus on tasks such as sentiment classification, syntactic/semantic tags prediction, NLI, etc.

However, there has not been much work done to analyze BERT for complex tasks like RCQA. It is a challenging task because of 1) BERT's sheer number of parameters and non-linearity, 2) BERT does not have pre-defined roles across layers as compared to pre-BERT models like BiDAF, DCN, etc. for such complex tasks. In this section, various experiments are performed to try to map BERT's layers to the following functions, deemed necessary in previous models to reach the answer: (i) learn contextual representations for the passage and the question, individually (ii) attend to information in the passage specific to the question (iii) predict the answer.

2.8.1 Explaining JSD

As described in Section 2.2.3, a layer's function is quantified and visualized as how it distributes importance over the passage words, using the distribution I_l. To compute the similarity between any two layers x, y, *Jensen-Shannon Divergence (JSD)* is measured between their corresponding importance distributions I_x, I_y. The JSD scores are calculated between every pair of layers in the model, and are visualised as a $n_l \times n_l$ heatmap (n_l is the number of layers in the model). Higher JSD score corresponds to the two layers being more different. This further means the two layers consider different words as salient. All heatmaps visualised in this section are the experiment-corresponding heatmaps averaged over 500 samples in SQuAD's dev-split.

2.8.2 Overall JSD analysis

Figure 2.6 shows pairwise layer JSD scores for BERT. Low JSD scores are observed between all pairs of layers, with only minimal increase as the layers go further apart(min/max JSD observed is just 0.06/0.41) giving a preliminary result that the layers are highly similar to each other.

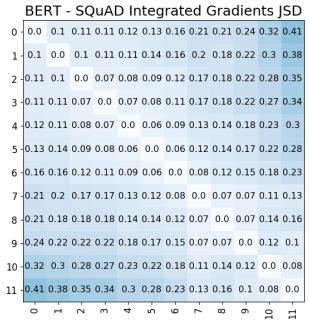
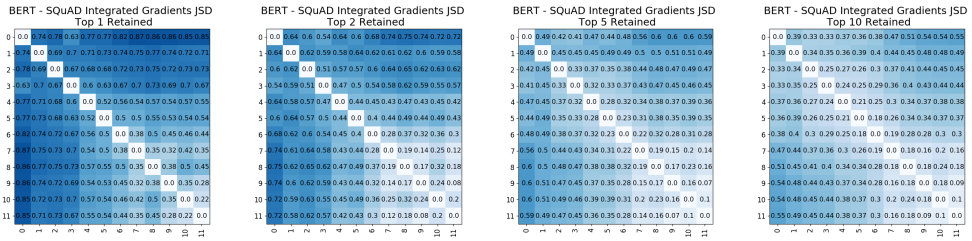


Figure 2.6: JSD between I_l's

2.8.3 JSD with top-k retained/removed

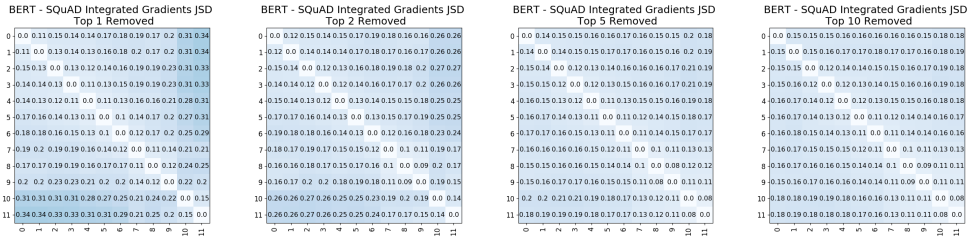
To further evaluate the source of the similarity, the I_l's are analysed in two parts: (i) only top-k scores retained in each layer and the rest zeroed out. This denotes the head of the distribution. (ii) top-k scores zeroed out in each layer and the rest retained, which denotes the tail of the distribution. In either case the distribution is re-normalized to

maintain the probability sum. The resulting heatmaps can be seen in Figures 2.7, 2.8.



(a) (b) (c) (d)

Figure 2.7: JSD between I_l's with only top-k items retained



(a) (b) (c) (d)

Figure 2.8: JSD between I_l's with top-k items removed

When comparing just the top-2 items (heatmap 2.7b), higher values(min 0.08/max 0.72) are seen as compared to when the remaining items are compared in heatmap 2.8b (min 0.09/max 0.26). Further, for both the head and the tail, it can be seen that, as k increases (from 1-10), the JSD scores decrease. Hence it can be concluded that a layer's function is reflected in words high up in the importance distribution. As they are removed, there is an almost uniform distribution across the less important words. Hence to correctly identify a layer's functionality, only on the head(top-k words) need to be focused on, and not the tail.

2.8.4 JSD analysis split by question type

In this section, the JSD heatmaps(top-2 retained) are analyzed split by question type, on the motivation that the model approaches different question types differently. For example, "what" or "who" questions require entities as answers, and in SQuAD can probably be answered more directly, whereas questions like "why" or "how" require a more in-depth reading of the passage. The results can be found in Figure 2.9.

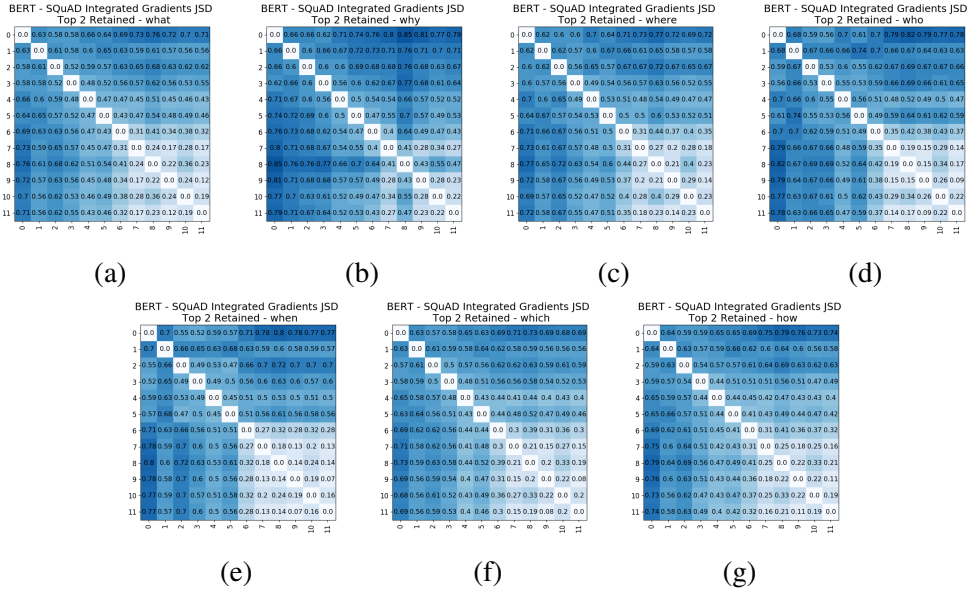


Figure 2.9: JSD of I_l's, split by question types

All heatmaps except the one for 'why' indicate behaviour similar to that observed in previous sections (for example consider the 'what' heatmap 2.9a). However, the heatmap for "why" (Fig. 2.9b), shows a slightly higher JSD in the later layers as well, supporting the hypothesis that such questions require a deeper understanding of the passage and hence more work to be done by the model.

2.8.5 Probing Layer : QA Functionality

Based on their functionality I_l, the layers are analyzed to see which of them focus more on the question, the context around the answer, etc. The passage words are segregated into three categories: *answer words, supporting words, query words*, where supporting words are the words surrounding the answer(+/- 5 words), and query words are the question words which appear in the passage. Following the discussion in the JSD experiment, the top-5 words marked as important in I_l are taken to represent the layer l. Then the percentage of words from each of the above-defined categories in the top-5 words is calculated. These results are shown in Table 2.7.

From Column 3, it is evident that the model first tries to identify the part of the passage where the question words are present, and as it gets more confident about the answer(Column 2), the importance of the question words decreases. From Column 4, we infer that it starts paying more importance to the support words, once the question

Layer Name	% answer span	% Q-words	% Contextual Words
Layer 0	26.99	22.94	9.45
Layer 1	26.09	24.35	9.43
Layer 2	29.9	22.41	11.65
Layer 3	30.44	19.55	11.13
Layer 4	30.06	18.33	11.23
Layer 5	30.75	14.71	11.57
Layer 6	31.25	15.33	11.94
Layer 7	32.37	12.29	12.32
Layer 8	30.78	18.91	12.07
Layer 9	34.58	10.21	13.41
Layer 10	34.31	10.56	13.39
Layer 11	34.63	12.0	13.74

Table 2.7: Semantic statistics about top-5 words

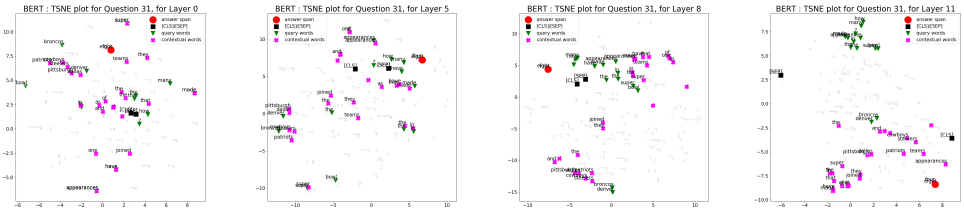
is understood, complementing the observation about Column 3.

Question: Why was Polonia relegated from the country's top flight in 2013? **Answer:** disastrous financial situation			
L0	Polonia was relegated from the country's top flight in 2013 because of their disastrous financial situation. They are now playing in the 4th league....	**L9**	Polonia was relegated from the country's top flight in 2013 because of their disastrous financial situation. They are now playing in the 4th league....
L1	Polonia was relegated from the country's top flight in 2013 because of their disastrous financial situation. They are now playing in the 4th league....	**L10**	Polonia was relegated from the country's top flight in 2013 because of their disastrous financial situation. They are now playing in the 4th league....
L2	Polonia was relegated from the country's top flight in 2013 because of their disastrous financial situation. They are now playing in the 4th league....	**L11**	Polonia was relegated from the country's top flight in 2013 because of their disastrous financial situation. They are now playing in the 4th league....

Table 2.8: Sample I_l over BERT's first and last 3 layers, visualised as a heatmap

.

Qualitative Example: Table 2.8 shows a visualization of the top-5 words of the first and last three layers(with respect to I_l). It is seen that all these six layers give a high score to the answer span itself ('disastrous', 'situation'). Further, the initial layers 0,1 and 2 are also trying to make a connection between the passage and the query ('relegated', 'because', 'Polonia' get high importance scores). Hence, in this example, it can be seen that the initial layers incorporate interaction between the query and passage. In contrast, the last layers focus on enhancing and verifying the model's prediction.



(a) (b) (c) (d)

Figure 2.10: T-sne plots across Layer 0, 5, 8, 11 to analyse how the word representations evolve across layers.

2.8.6 Visualizing Word Representations

In this section, the word representations learnt by each layer are analyzed qualitatively. The t-SNE plot for one such passage, question,answer triplet (refer Table 2.9) is visualized in Figure 2.10. The answer, supporting words, query words, and special tokens are visualized. Note that the other words in the passage have been grayed out.

Passage: the panthers finished the regular season with a 15 – 1 record, ... the broncos ... finished the regular season with a 12 – 4 record. They joined the patriots , dallas cowboys , and pittsburgh steelers as one of teams that have made **eight** appearances in the super bowl .
Question: How many appearances have the Broncos made in the super bowl?

Table 2.9: Sample from the dev-split of SQuAD. **Blue** shows the answer, purple shows the contextual passage words and **green** shows the query

.

In initial layers (such as layer 0), similar words are closer to each other: such as stopwords, team names, numbers {eight, four} etc. From Layer 5 onwards, the passage, question, and answer come close to each other. By layer 8, the answer words are segregated from the rest of the words, even though the passage word 'four' which is of the same type as the answer 'eight'(number) is still close to 'eight'. More interesting observations yet here: (i) in later layers the question words separate from the answer and the supporting words, (ii) Across all 12 layers, embeddings for *four, eight* remain very close together, which could have easily led to the model making a wrong prediction. However, the model still predicts the answer 'eight' correctly; it is not possible to identify the layer where the distinction between the two confusing answers occurs.

Quantifier questions: Further detailed analysis is done on quantifier questions like

how many, how much that could potentially have many confusing answers(i.e., numerical words) in the passage. Based on our layer-level functionality I_l, the number of words that are numerical quantities in the top-5 words, and in the entire passage are computed and their ratio is obtained. This represents the ratio of confusing words marked as important by each layer. Interestingly, it is observed that this ratio *increases* as from the intial to final layers (5% at layer 0 to 18% layer 11). For this example in particular, in its later layers, BERT gives high importance to the words 'eight', 'four', and 'second' (numerical quantities), even though the latter is not necessary or related to answer the question.

This shows that BERT, in its later layers, distributes its importance over potentially confusing words; however, it still manages to predict the correct answer for such questions (87.42% EM for how much/many questions). This behavior is very different from the assumed roles a layer might take to answer the question; it would have been expected that such words were considered in the initial rather than final layers. This observation shows the complexity of BERT and the difficulty of interpreting for an end goal like QA.

2.9 Conclusion

In this project, four deep QA models were analyzed for interpretability at the layer-level and the model-level using integrated gradients. Based on the definition for model explainability given by Serrano and Smith (2019), it was found that the models DCN, BiDAF, BERT, and QANET could be classified as explainable. However, based on human evaluation and qualitative analysis of extracted rationale, it was found that though the models are comparable to humans in terms of performance, there is a wide gap in the rationale given for its prediction. Further, for comparison purposes, a short discussion was provided on two other interpretability methods: LIME and inbuilt attention mechanisms. The models' embeddings across layers were analyzed quantitatively and qualitatively using conicity (and tSNE plots for BERT). Finally, results and analysis were presented to understand the QA-specific roles of BERT's various layers; it was observed that BERT was learning some form of passage-query interaction in its initial layers before arriving at the answer. Other interesting trends were also observed and discussed in the experiments.

2.10 Future Work

Other experiments done in this work include pruning BERT's layers to analyze their contribution to performance, JSD heatmaps for BiDAF/QANet/DCN, and preliminary analysis of BERT for the more complex dataset DuoRC (Saha *et al.*, 2018). These results being preliminary and away from the central theme of the project, are included in the appendix (refer to A). As future work, these precursory experiments can be explored to understand the models better. Further, this analysis can be extended to multihop datasets such as WikiHopWelbl *et al.* (2018), HotPotQAYang *et al.* (2018) etc. as well to more QA models.

CHAPTER 3

DIALOG-BASED IMAGE RETRIEVAL

3.1 Dataset

Multiple datasets such as VisDial Das *et al.* (2017*a*), CLEVR Dialog Kottur *et al.* (2019) and Fashion-IQ Guo *et al.* (2019) exist for the task of image retrieval through multi-modal conversation. In this project, CLEVR Dialog is the dataset used. CLEVR Dialog is a synthetically created dataset where the images and dialogs have restricted domains; the dataset has a small number of object types, attribute values, and relations. These properties provide a simplistic setting in which the model's working(abilities, biases, and limitations) can be thoroughly analyzed.

3.2 Task Setting

The task (Figure 3.1) involves a conversation between two agents - the questioner(QBot) and the answerer(ABot); at the end, the QBot has to accumulate information from the conversation and retrieve the matching image from the catalog of images.

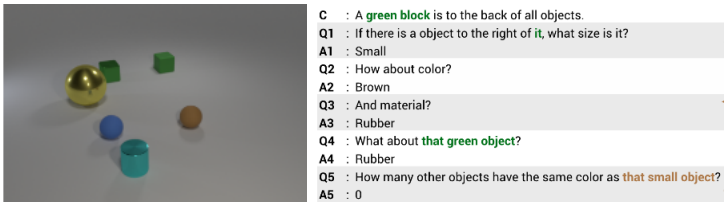


Figure 3.1: CLEVR Dialog Dataset

1. A-Bot picks **image** from catalog.
2. A-Bot gives **caption** about it.
3. Q-Bot hallucinates image, asks **question**.
4. A-Bot gives **answer**.
5. Steps 3,4 repeat till Q-Bot **retrieves** image.

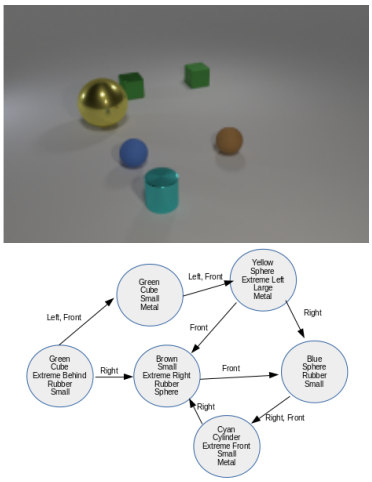
In this work, the aim is two-way modelling of the Q-Bot and the A-Bot, posing the image retrieval as a strategic search problem where the goal is reached through multiple iterations of the Q-Bot and A-Bot.

3.3 Traditional Multimodal Conversation Architecture

Most papers published on these datasets model **only** the answerer bot. Effectively, the task reduces to answering a natural language question based on an image. The few works which do model the questioner bot(for example the work *Learning Cooperative Agents using Deep RL* Das *et al.* (2017*b*)), use a fully neural model, working completely in the image and text vector spaces. The A-Bot combines the representation of the question and the image to give the answer; the Q-Bot encodes the entire conversation into a single vector and projects it into the image vector space, which is then matched to an image in the catalog on the basis of closest euclidean distance. However, such an end-to-end neural model fails to explicitly model the latent structures present in the two modalities and the various strategies required in this task.

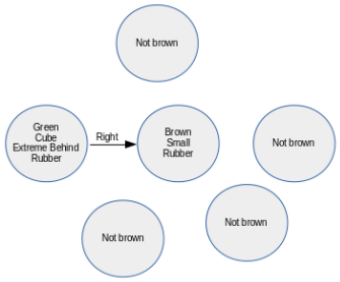
The goal of this project is to design a neural symbolic model which explicitly uses scene graphs to represent the dialog and the catalog of images; strategies and structural constraints of the task are captured in the neural-symbolic reasoning between the multimodal graphical representations.This model also explicitly uses the catalog of images to ask the next question, rather than ignoring it as done by existing works.

3.4 Scene Graphs

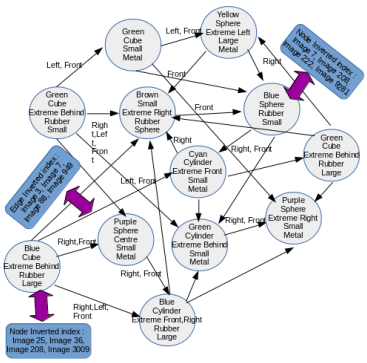


(a) Image's **gold** scene graph

C : A green block is to the back of all objects.
Q1 : If there is a object to the right of it, what size is it?
A1 : Small
Q2 : How about color?
A2 : Brown
Q3 : And material?
A3 : Rubber
Q4 : What about that green object?
A4 : Rubber
Q5 : How many other objects have the same color as that small object?
A5 : 0



(b) Dialog's partial **query** scene graph



(c) Global **Catalog** scene graph

Figure 3.2: Scene Graph Representations

The individual images, the ongoing dialog and the catalog of images are all represented as scene graphs (refer Figure 3.2), with the objects in the image as nodes in the scene graph and spatial relations between them as the edges. The catalog scene graph is constructed by merging the scene graphs of all the images in the catalog; an inverted index is also constructed to match each node, edge and head-edge-tail triplet with all the images it is present in.

The scene graphs of the images can be constructed by applying models such as Faster R-CNN (Ren *et al.*, 2015), and taking the bounding boxes as nodes and spatial relations between these boxes as edges. The scene graph of individual sentences can be constructed using the SPICE framework (Anderson *et al.*, 2016), which uses the dependency structure of the text followed by multiple tree transformations to get the final

scene graph of the text.

A consolidated catalog scene graph is used instead of multiple separate scene graphs of the catalog images for 2 reasons. Firstly, it is easier to match a partial query graph to subgraphs of a single, huge graph rather than matching it to multiple (relatively) bigger graphs. Secondly, from the perspective of the Q-Bot, it is easier to ask the next question based on a single graph, rather than aggregating it from multiple smaller graphs during runtime.

3.5 One-Shot Image Retrieval

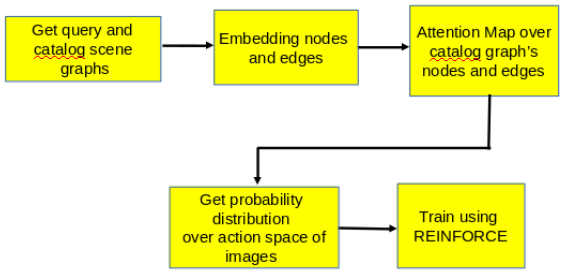


Figure 3.3: Pipeline for One Shot Image Retrieval

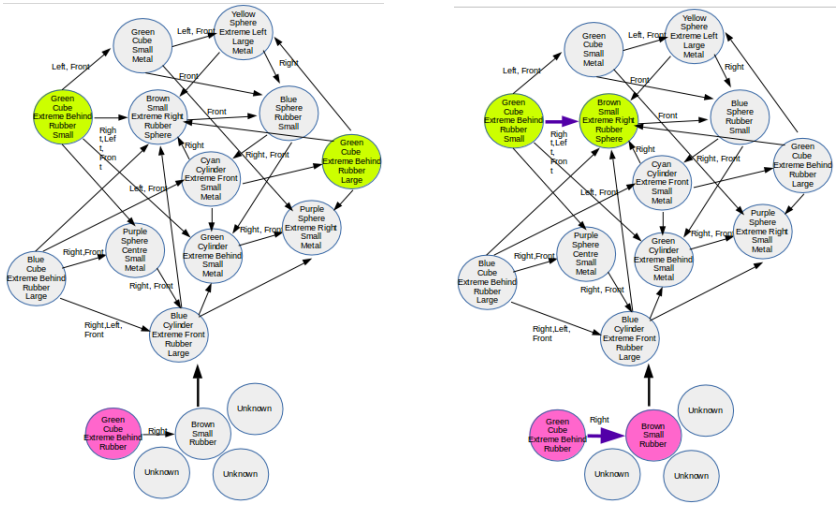
EMBEDDING NODES AND HEAD-EDGE-TAIL TRIPLES

Each node is represented as a list of 6 fixed attributes : [shape, colour, size, material, uniqueness, extremeness] where each of these can take on a fixed number of values only.

Each triple is represented as a list of the head and tail node attributes (from above), and the relation type of the edge.

The node/triple is represented as a matrix $M \times N$ of the embeddings of its attributes; currently, GloVe embeddings (Pennington *et al.*, 2014) are used, however, the final model can use multimodal embeddings as obtained from the bounding boxes representing the nodes.

ATTENTION MAP OVER NODES AND TRIPLES



(a) Attention map over nodes (b) Attention map over triples

Figure 3.4: Attention Maps

An attention map is calculated over the catalog graph's nodes/triples for each node/triple in the query graph using the **frobenius norm** between their corresponding matrices of embeddings ($\frac{1}{1+d}$). These attention maps are used to calculate action probabilities as discussed below.

ACTION SPACE FOR IMAGE RETRIEVAL

The **action space** is defined as all the **images** which the Q-Bot has access to.

$$P(\text{action=image}) = \prod_{\text{query nodes}} max(\text{score catalog nodes/edges} \in \text{image})$$

In this way, each image is scored according to its relevance to each query node and triple. This ensures that the best image(s) according to **all** the query entities are selected.

TRAINING USING REINFORCE

Training is done in a RL setting with the images as the actions and the policy as obtained above.

$$\theta_{t+1} = \theta_t + \alpha \sum_{images} P(\text{image})[R(\text{image}) - B] \log(P(\text{image}))$$

Weights are updated by increasing the REINFORCE reward, which is reducing euclidean distance between the ResNet (He *et al.*, 2016) representations of the predicted and the gold image. Baseline is a running average over rewards.

3.5.1 Experiments - One Shot Retrieval

The one-shot retrieval model is tested on complete as well as simulated partial scene graphs from the CLEVR dataset.

Setting	On average	Retrieval Accuracy
With CLEVR gold scene graphs	6 nodes, 6 attributes, 20 edges	$\simeq 99.9\%$
Simulated partial scene graphs (20% entities dropped)	5 nodes, 5 attributes, 18 edges	$\simeq 89\%$
Simulated partial scene graphs (30% entities dropped)	4 nodes, 4 attributes, 15 edges	$\simeq 75\%$

(Note that these numbers are for exact match of retrieved image with the target image).

With the one-shot retrieval pipeline as the backbone, an iterative retrieval process is built for the Q-Bot and A-Bot.

3.6 Full Iterative dialog-based retrieval

The full iterative retrieval pipeline can be seen in Figure 3.5. The training for these modules over the full dialog :

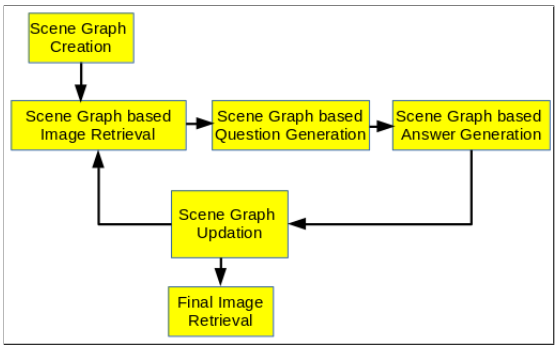


Figure 3.5: Full Retrieval Pipeline

- At the 0^{th} round, scene graph based image retrieval is done
- At the ith round :
  - Q-Bot samples from action space, generates question
  - A-Bot answers the question
  - Q-Bot updates query graph based on the answer
  - Scene graph based image retrieval is done
  - Modules are trained end-to-end based on retrieval reward

3.6.1 Action Space for QBot

The action taken by the QBot (question asked) must satisfy two criteria : it must add new information to the query graph and at any point in the conversation, it must be the question that adds the most information to the query graph. That is, it needs to add valuable and non-redundant information to the query graph. Keeping this in mind, two types of actions are made available to the QBot in the full pipeline : **node attribute**(pick a node and ask for an unknown attribute value) and **edge-value** (pick a node and ask about the presence/absence of its edges).

Both these actions are termed as *categorical* actions, and are considered as representative of multiple *boolean* actions. For example, asking what the colour of a node is, is a superset of asking whether the node has a colour of blue,red,gray,green etc. Similarly, asking if a node has any node to its left is the superset of asking if its left edge is present or absent.

EMBEDDING BOOLEAN ACTIONS

For each query node, multiple attribute boolean actions are created, each represented as a list of the query node's known attributes, one unknown attribute replaced with a wildcard value and the remaining unknown values retained as such. Similarly, multiple edge boolean actions are created, each as a list of known edges and one unknown edge replaced with either *edge-value* or *none*. Both of these are then correspondingly translated into an embedding matrix, similar to node/edge embeddings.

ATTENTION MAP FOR ACTIONS

Treating each boolean action as similar to a node/edge, an attention map is computed

over the catalog graphs nodes and edges.

PROBABILITY OF ACTIONS

Next, each boolean action gets an action score as the product of attention over the catalog graph with respect to the actions as well as the query graph. Further, each node/edge in the catalog graph is assigned a tally of how many images they are a part of. This tally is multiplied with the above obtained score, is summed across catalog nodes/edges, and then normalized to get a probability or **shatter score** for each action. This shatter score is a measure of how well the each boolean action splits the image space.

$$action\_score = attn(action) \times attn(query\ node/edge)$$
$$shatter\_score = \sum_{catalog} action\_score \times tally$$

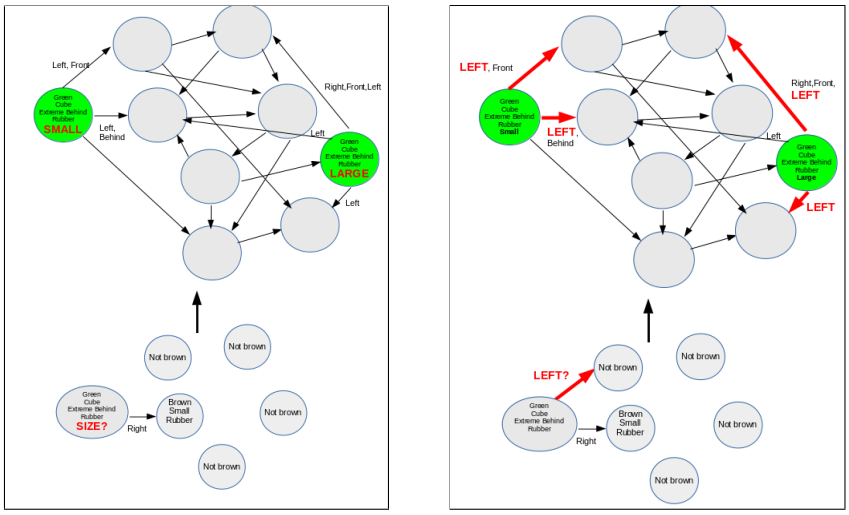
ENTROPY OF ACTIONS

Finally, entropy is calculated across these boolean action scores to get a score for each categorical action. The QBot selects the categorical action which has the *minimum* entropy, helping to narrow down the image space as much as possible at each round of conversation.

$$categorical\_score = entropy(boolean)$$
$$\text{action picked} \leftarrow argmin(categorical\_score)$$

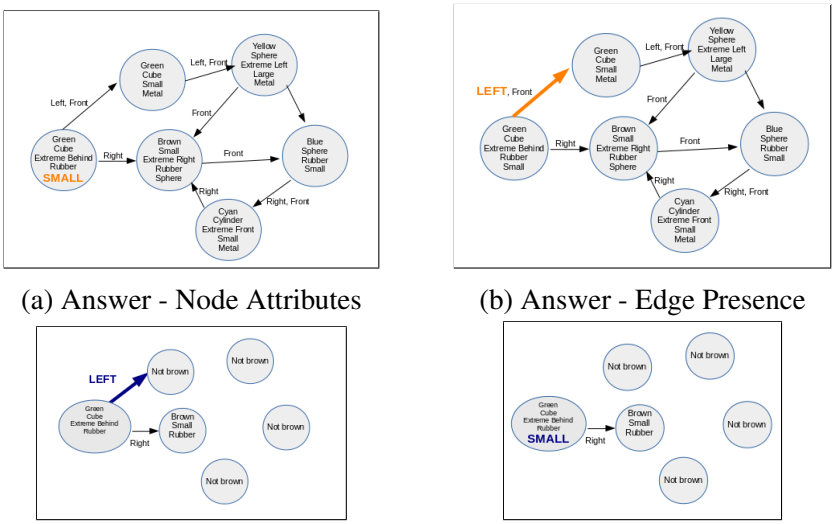
3.6.2 Joint functioning of QBot and ABot

The QBot picks an action and sends it to the ABot. The ABot compares the question and node asked to its gold image scene graph and returns an appropriate answer. The QBot then uses this answer and updates its query scene graph accordingly. This joint functioning of the 2 agents is illustrated in Figures 3.6 and 3.7.



(a) Question on Node Attributes (b) Question on Edge Presence

Figure 3.6: Possible actions for Q-Bot



(a) Answer - Node Attributes (b) Answer - Edge Presence

(c) Updation - Node Attributes (d) Updation - Edge Presence

Figure 3.7: Answer and Scene Graph Updation

3.6.3 Experiments - Iterative Image Retrieval

The full iterative retrieval model is tested on the validation set of CLEVR Dialog dataset. However, instead of starting from the caption graph which is too sparse for usage by this model, the starting query graph is the scene graph at the end of the dialogs in the dataset, which by themselves are fairly sparse. The model is tested with varying rounds of conversation and results can be found in Table (Note that these numbers are for exact match of retrieved image with the target image)

Question Types	5 rounds	10 rounds	20 rounds	40 rounds
Node Attributes	27.4	61.6	83.06	-
Node Attributes, Edge Type	8.12	15.3	48.89	88.32

Table 3.1: Image retrieval accuracy (%) after iterative rounds of QBot-ABot conversation, using dialog gold query graphs as input and node-attribute and and edge-type based questions. QBot uses full catalog graph to ask questions. This is for 75000 datapoints from the CLEVR dialog validation datasets.

3.7 Conclusion and Future Work

In this project, a novel neural-symbolic iterative image retrieval model was introduced and implemented, and competitive results were shown on the CLEVR and CLEVR Dialog datasets.

As future work, this iterative pipeline has to be modified to use probabilistic scene graphs generated from the image, rather than exact scene graphs present in the dataset itself. Existing works such as NS-VQA (Yi *et al.*, 2018), NSCL (Mao *et al.*, 2019) etc. can be used for this purpose. Further, more complicated actions such as triplet questions (for example, *what is the color of the node to the left of node 1?*), and higher-subgraph questions(*How many yellow objects are present in this image?, How many nodes are present to the right of node 2?*) can be implemented to increase the power of the model. Finally, once the model is perfected on the CLEVR Dialog dataset, it can be extended to the more difficult natural datasets such as VisDial.

APPENDIX A

ANALYZING INTERPRETABILITY OF DEEP RCQA SYSTEMS

A.1 BERT - Pruning Layers

Pruning experiments on BERT involve the removal of certain layers from the original pre-trained BERT, followed by training on SQuAD. The subsequent change in performance is shown in Table A.1). The layers are dropped iteratively to identify the impact they have on the model's performance (first layer 11 dropped, then 10 & 11 dropped and so on until layer 6).

It is seen that pruning layer 11 causes almost no change in the performance(only a

Layers pruned	%F1	%Drop in F1
None	88.73	-
11	88.66	0.07
10,11	87.81	0.85
9,10,11	86.58	1.23
8,9,10,11	86.4	0.18
7,8,9,10,11	85.15	1.25
6,7,8,9,10,11	83.75	1.4

Table A.1: Pruned BERT models on SQuAD's dev-set

0.07% dip). Dropping layers 10 and then 9 cause a further dip of $\sim$1% each. However, dropping layer 8 does not have a huge impact(only 0.18%). Again, dropping layers 7 and then 6 have a tangible impact(1.25% and 1.4% respectively). Removing layers 11 and 8 cause almost no dip to the model's performance; perhaps they are redundant. However removal of layers 10,9,7 cause a noticeable dip in performance; preliminarily, it can be said that they either contribute to the model's logic, or perform necessary mathematical perturbations in the model's high dimensions.

A.2 JSD plots for BiDAF, DCN, QANet

The following figures contain the overall JSD heatmaps as well as top-k retained/removed JSD heatmaps for BiDAF, DCN and QANet.

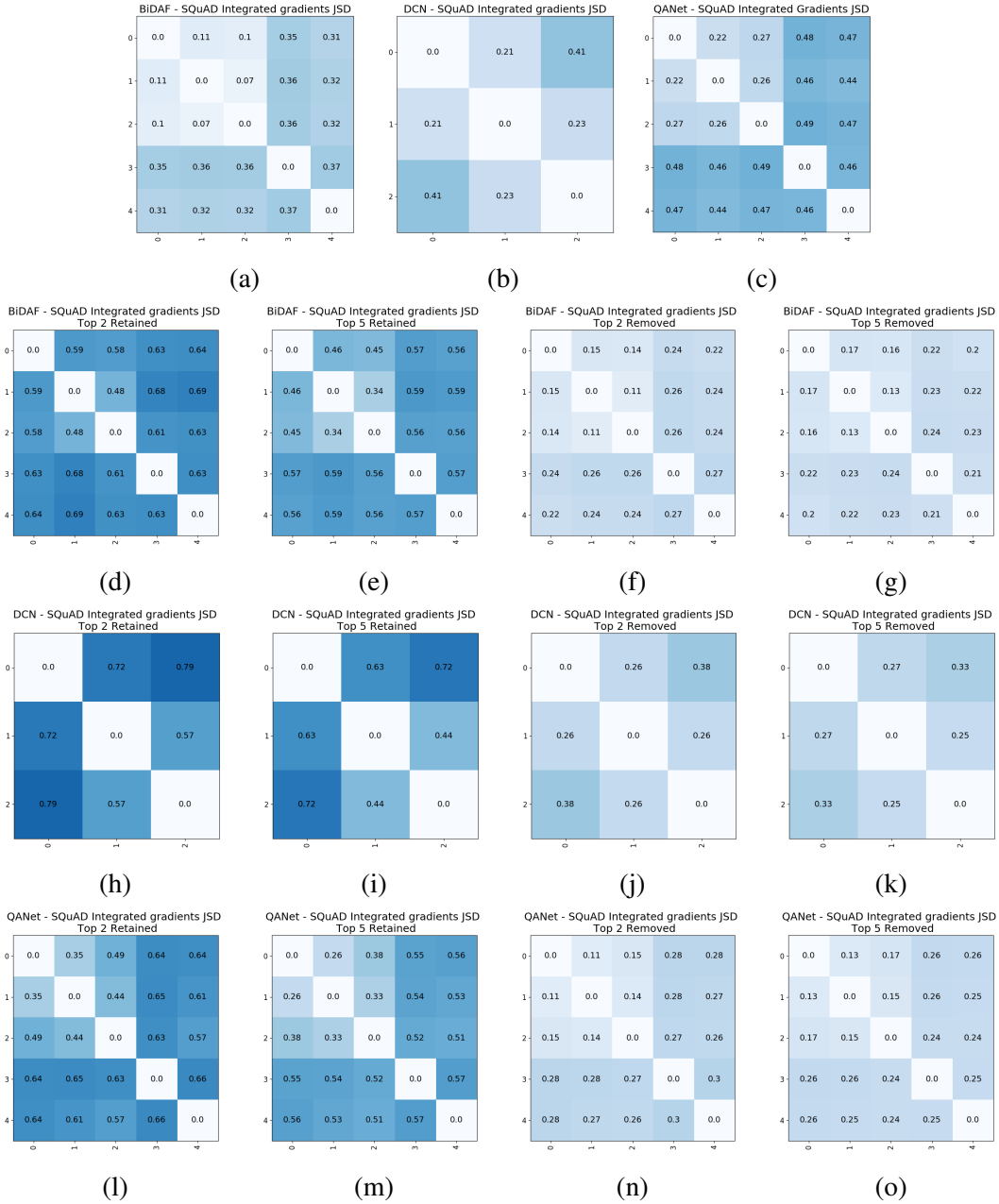
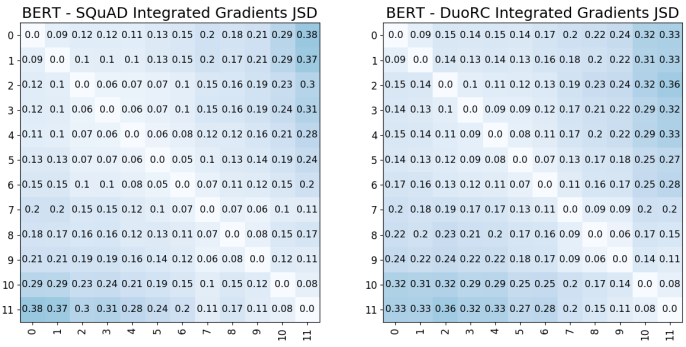


Figure A.1: JSD of importance score distributions overall and with top-k items removed/retained, averaged across 500 datapoints of SQuAD for BiDAF, DCN and QANet

A.3 BERT on DuoRC

This section analyzes the interpretability of QA models, specifically BERT on the more complex dataset DuoRC Saha *et al.* (2018), which is explicitly created in a way that arriving at the answer requires one to focus on different sentences in the passage. The goal here is to examine if the model highlights more relevant words in such datasets where humans cannot infer the answer by looking at a small neighborhood around the answer span. After fine-tuning, BERT achieved an F1 score of 54.9 on DuoRC, which is comparable to the state-of-the-art performance for this dataset. Similar to the results in Sections 2.4 and 2.7, it was observed that even for DuoRC where elaborate steps are needed to reach the answer, BERT still somehow directly learns to focus on the answer span, [CLS], and [SEP] tokens (even in the early layers). Further, it is found that the JSD scores for this dataset follow a similar trend as that observed on the SQuAD dataset given in Figure A.2.



(a) (b)

Figure A.2: JSD of integrated gradients scores, averaged across 500 datapoints for (a) BERT on SQuAD (b) BERT on DuoRC

REFERENCES

1. **Anderson, P.**, **B. Fernando**, **M. Johnson**, and **S. Gould**, Spice: Semantic propositional image caption evaluation. *In European Conference on Computer Vision*. Springer, 2016.

2. **Bach, S.**, **A. Binder**, **G. Montavon**, **F. Klauschen**, **K.-R. Müller**, and **W. Samek** (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, **10**(7), e0130140.

3. **Clark, K.**, **U. Khandelwal**, **O. Levy**, and **C. D. Manning** (2019). What does BERT look at? an analysis of bert's attention. *CoRR*, **abs/1906.04341**. URL http://arxiv.org/abs/1906.04341.

4. **Das, A.**, **S. Kottur**, **K. Gupta**, **A. Singh**, **D. Yadav**, **J. M. Moura**, **D. Parikh**, and **D. Batra**, Visual dialog. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017*a*.

5. **Das, A.**, **S. Kottur**, **J. M. Moura**, **S. Lee**, and **D. Batra**, Learning cooperative visual dialog agents with deep reinforcement learning. *In Proceedings of the IEEE International Conference on Computer Vision*. 2017*b*.

6. **Devlin, J.**, **M. Chang**, **K. Lee**, and **K. Toutanova** (2018*a*). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, **abs/1810.04805**. URL http://arxiv.org/abs/1810.04805.

7. **Devlin, J.**, **M.-W. Chang**, **K. Lee**, and **K. Toutanova** (2018*b*). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

8. **Guo, X.**, **H. Wu**, **Y. Gao**, **S. Rennie**, and **R. Feris** (2019). The fashion iq dataset: Retrieving images by combining side information and relative natural language feedback. *arXiv preprint arXiv:1905.12794*.

9. **He, K.**, **X. Zhang**, **S. Ren**, and **J. Sun**, Deep residual learning for image recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

10. **Jia, R.** and **P. Liang** (2017). Adversarial examples for evaluating reading comprehension systems. *CoRR*, **abs/1707.07328**. URL http://arxiv.org/abs/1707.07328.

11. **Kottur, S.**, **J. M. Moura**, **D. Parikh**, **D. Batra**, and **M. Rohrbach** (2019). Clevr-dialog: A diagnostic dataset for multi-round reasoning in visual dialog. *arXiv preprint arXiv:1903.03166*.

12. **Lan, Z.**, **M. Chen**, **S. Goodman**, **K. Gimpel**, **P. Sharma**, and **R. Soricut** (2019). Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

13. **Lei, T.**, **R. Barzilay**, and **T. Jaakkola** (2016). Rationalizing neural predictions. *arXiv preprint arXiv:1606.04155*.

14. **Mao, J.**, **C. Gan**, **P. Kohli**, **J. B. Tenenbaum**, and **J. Wu** (2019). The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *arXiv preprint arXiv:1904.12584*.

15. **Pennington, J.**, **R. Socher**, and **C. Manning**, Glove: Global vectors for word representation. *In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014.

16. **Peters, M. E.**, **M. Neumann**, **L. Zettlemoyer**, and **W.-t. Yih** (2018). Dissecting contextual word embeddings: Architecture and representation. *arXiv preprint arXiv:1808.08949*.

17. **Rajpurkar, P.**, **J. Zhang**, **K. Lopyrev**, and **P. Liang** (2016). Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

18. **Ren, S.**, **K. He**, **R. Girshick**, and **J. Sun**, Faster r-cnn: Towards real-time object detection with region proposal networks. *In Advances in neural information processing systems*. 2015.

19. **Ribeiro, M. T.**, **S. Singh**, and **C. Guestrin**, Why should i trust you?: Explaining the predictions of any classifier. *In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 2016.

20. **Saha, A.**, **R. Aralikatte**, **M. M. Khapra**, and **K. Sankaranarayanan** (2018). Duorc: Towards complex language understanding with paraphrased reading comprehension. *CoRR*, **abs/1804.07927**. URL http://arxiv.org/abs/1804.07927.

21. **Seo, M.**, **A. Kembhavi**, **A. Farhadi**, and **H. Hajishirzi** (2016*a*). Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

22. **Seo, M. J.**, **A. Kembhavi**, **A. Farhadi**, and **H. Hajishirzi** (2016*b*). Bidirectional attention flow for machine comprehension. *CoRR*, **abs/1611.01603**. URL http://arxiv.org/abs/1611.01603.

23. **Serrano, S.** and **N. A. Smith** (2019). Is attention interpretable? *arXiv preprint arXiv:1906.03731*.

24. **Sharma, A.**, **P. Talukdar**, *et al.*, Towards understanding the geometry of knowledge graph embeddings. *In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2018.

25. **Shrikumar, A.**, **P. Greenside**, and **A. Kundaje**, Learning important features through propagating activation differences. *In Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017.

26. **Si, C.**, **S. Wang**, **M.-Y. Kan**, and **J. Jiang** (2019). What does bert learn from multiple-choice reading comprehension datasets? *arXiv preprint arXiv:1910.12391*.

27. **Sundararajan, M.**, **A. Taly**, and **Q. Yan**, Axiomatic attribution for deep networks. *In Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017.

28. **Sutton, R. S.**, **D. A. McAllester**, **S. P. Singh**, and **Y. Mansour**, Policy gradient methods for reinforcement learning with function approximation. *In Advances in neural information processing systems*. 2000.

29. **Tenney, I.**, **D. Das**, and **E. Pavlick** (2019). Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*.

30. **Welbl, J.**, **P. Stenetorp**, and **S. Riedel** (2018). Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, **6**, 287–302.

31. **Xiong, C.**, **V. Zhong**, and **R. Socher** (2016*a*). Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604*.

32. **Xiong, C.**, **V. Zhong**, and **R. Socher** (2016*b*). Dynamic coattention networks for question answering. *CoRR*, **abs/1611.01604**. URL http://arxiv.org/abs/1611.01604.

33. **Yang, Z.**, **P. Qi**, **S. Zhang**, **Y. Bengio**, **W. W. Cohen**, **R. Salakhutdinov**, and **C. D. Manning** (2018). Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

34. **Yi, K.**, **J. Wu**, **C. Gan**, **A. Torralba**, **P. Kohli**, and **J. Tenenbaum**, Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. *In Advances in Neural Information Processing Systems*. 2018.

35. **Yu, A. W.**, **D. Dohan**, **M. Luong**, **R. Zhao**, **K. Chen**, **M. Norouzi**, and **Q. V. Le** (2018*a*). Qanet: Combining local convolution with global self-attention for reading comprehension. *CoRR*, **abs/1804.09541**. URL http://arxiv.org/abs/1804.09541.

36. **Yu, A. W.**, **D. Dohan**, **M.-T. Luong**, **R. Zhao**, **K. Chen**, **M. Norouzi**, and **Q. V. Le** (2018*b*). Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*.

LIST OF PAPERS BASED ON THESIS

1. Scene Graph based Image Retrieval – A case study on the CLEVR Dataset - extended abstract awarded the Best Paper at LINGIR Workshop, ICCV 2019.
2. Submission at ACL 2020 (weak reject)
3. Two anonymous short paper submissions at EMNLP 2020.